

Steganografi Teks pada Aksara Sunda dengan Pendekatan *Feature Coding*

Mohammad Rijal, Henning Titi Ciptaningtyas, dan Tohari Ahmad

Departemen Teknik Informatika, Fakultas Teknologi Informasi, Institut Teknologi Sepuluh Nopember (ITS)

e-mail: henning@if.its.ac.id

Abstrak—Kemajuan teknologi membuat arus informasi semakin meningkat. Kebutuhan akan transmisi data yang aman menjadi sangat penting. Dalam dunia sekuritas, dikenal istilah steganografi. Steganografi merupakan teknik untuk menyembunyikan informasi tanpa diketahui keberadaan informasi rahasia tersebut. Salah satu jenis steganografi adalah steganografi teks yang menggunakan teks sebagai media penyampaian. Aksara Sunda yang memiliki 13 rarangkén atau vokalikasi dapat digunakan sebagai media dalam steganografi teks. Metode yang ditawarkan pada penelitian ini menggunakan pendekatan *Feature Coding* pada aksara Sunda. *Feature Coding* mengubah bagian dari teks yang menjadi media untuk menyimpan bit informasi. Pada aksara Sunda, rarangkén digeser untuk menyimpan bit *secret message* yang menjadi masukan. Pilihan pergeseran pada penelitian ini terbagi menjadi *shift all* yang menggeser semua rarangkén dan *shift group* yang menggeser kelompok posisi rarangkén dalam satu huruf. Hasil uji coba metode menunjukkan bahwa steganografi teks pada aksara Sunda berhasil dilakukan dengan menggeser rarangkén melalui penggantian blok Unicode. *Capacity ratio* yang dihasilkan dipengaruhi oleh jumlah rarangkén yang ada. Pilihan *shift all* menghasilkan rata-rata *capacity ratio* 57,01 bit/kiloByte. Pilihan *shift group* menghasilkan rata-rata *capacity ratio* yang lebih besar yaitu 60,74 bit/kiloByte. Kenaikan sebesar 6,54% disebabkan oleh jumlah rarangkén pada pilihan *shift group* yang lebih besar dari pilihan *shift all*.

Kata Kunci—Steganografi, Aksara Sunda, *Feature Coding*, *Secret Message*, *Capacity Ratio*, Rarangkén.

I. PENDAHULUAN

KEMAJUAN teknologi membuat arus informasi semakin meningkat. Kebutuhan akan transmisi data yang aman menjadi sangat penting. Dalam dunia sekuritas, dikenal istilah steganografi dan kriptografi. Steganografi merupakan suatu cara untuk menyembunyikan informasi tanpa diketahui keberadaan informasi rahasia tersebut. Media yang disisipi informasi rahasia dalam steganografi tak akan terlihat menampilkan perbedaan [1]. Berbeda dengan steganografi, kriptografi menampilkan pesan yang acak dalam penyembunyian informasi sehingga lebih mudah untuk diketahui keberadaannya [2].

Steganografi secara umum terbagi menjadi steganografi

gambar, audio, video, dan teks. Dari jenis tersebut, Steganografi Teks merupakan salah satu yang paling sulit disebabkan tingkat redundansi informasi yang rendah jika dibandingkan gambar atau audio [3]. Pada steganografi teks, penyembunyian informasi dapat digunakan pada semua bahasa. Shirali-Shahreza [4] memanfaatkan huruf yang mempunyai titik pada bahasa Arab untuk menyembunyikan bit informasi. Posisi titik pada huruf akan digeser sesuai dengan bit yang disembunyikan. Huruf bertitik dalam bahasa Arab yang berjumlah 13 membuat *hidden capacity ratio* yang dihasilkan tinggi. Penggunaan ekstensi dalam bahasa Arab dimanfaatkan oleh Gutub dan Fattani [5] untuk menyembunyikan bit informasi. Setiap huruf setelah huruf bertitik, penulisannya akan diberi ekstensi atau dipanjangkan. Metode ini mengatasi masalah robustness dari Shirali-Shahreza, tetapi terbatas pada kapasitas informasi karena tidak semua huruf Arab dapat mempunyai ekstensi. Aabed dkk [6] menggunakan Diakritik atau Vokalikasi yang bisa digunakan pada setiap huruf Arab. Ketiga metode tersebut merupakan metode dasar yang bervariasi pada perkembangannya. Variasi tersebut dapat dilihat dari *Kashida Variation Algorithm* dan *Multipoint Letter Algorithm* oleh Odeh [7] [8] yang memungkinkan penyimpanan 2 bit dalam satu huruf.

Steganografi teks yang menggunakan huruf non-Latin tidak terbatas pada huruf Arab. Alla dan Prasad [9] memanfaatkan diakritik dan *compound words* pada huruf Devanagari, huruf yang digunakan pada bahasa India. Sementara itu Changder dkk [10] menggunakan pegeseran Matra, vokalikasi yang digunakan pada konsonan dalam huruf Devanagari, untuk menyimpan bit informasi. Pada metode yang ditawarkan Alla dan Prasad [11], posisi matra yang berada di atas atau di bawah digunakan untuk menandai bit “0” atau “1”. Selain itu, Alla dan Prasad [12] mengembangkan metode steganografi dalam bahasa lokal di India yaitu Telugu dengan menggunakan kesesuaian *ottulu* dan tanda baca.

Aksara Sunda sebagai aksara Nusantara bersifat silabik, yaitu satu huruf mewakili nilai satu suku kata. Bunyi vokal dalam aksara Sunda dapat diganti dengan menggunakan rarangkén, yaitu vokalikasi atau diakritik yang ditulis di atas, di bawah, atau pun sejajar dengan huruf. [13] Dengan jumlah rarangkén yang mencapai 13 buah, aksara Sunda dapat digunakan sebagai alternatif media untuk steganografi teks.

Pada penelitian ini, penulis menggunakan Aksara Sunda

sebagai media untuk steganografi teks. Metode yang ditawarkan menggunakan pendekatan *Feature Coding*, di mana terdapat perubahan posisi dari rarangken setiap kali menyimpan bit informasi. Dengan jumlah rarangken yang tinggi dan hampir dapat dipakai pada setiap huruf, hal ini diharapkan dapat meningkatkan *capacity ratio* dari steganografi teks. Untuk menjaga *robustness*, setelah selesai disisipkan informasi, maka teks akan dikonversi ke dalam bentuk PDF.

II. METODE PENELITIAN

A. Deskripsi Umum Sistem

Sistem steganografi teks pada aksara Sunda dalam penelitian ini terbagi atas modul *embedding* dan *extracting*. Sistem menggunakan pendekatan *Feature Coding* melalui pergeseran posisi rarangken. Pada modul *embedding*, sistem menampilkan antarmuka untuk pengisian data masukan *secret message*, *cover text*, dan *shifting option*. Modul *embedding* bertujuan untuk melakukan proses penyembunyian informasi. Hasil keluaran modul *embedding* adalah *stego text* dalam bentuk PDF. Pada modul *extracting*, sistem menampilkan antarmuka untuk pengambilan berkas PDF dan pemilihan *shifting options*. Modul *extracting* melakukan ekstraksi *secret message* dari PDF. Hasil keluaran modul *extracting* adalah *secret message* yang dapat dilihat pada antarmuka program.

B. Deskripsi Modul Embedding

















Modul *embedding* bertujuan untuk melakukan proses penyembunyian pesan rahasia. Modul *embedding* menerima data masukan berupa *secret message*, *shifting option*, dan *cover text*. Proses dalam modul *embedding* secara umum terbagi menjadi empat tahap yaitu konversi *secret message*, pembacaan *cover text*, pergeseran rarangken, dan penyimpanan *stego text*.

Proses konversi *secret message* dimulai dengan mengubah *secret message* ke dalam nilai ASCII untuk setiap huruf. Kemudian, setiap nilai ASCII tersebut diubah ke dalam bentuk bit biner. Satu nilai ASCII direpresentasikan dengan 8 bit biner. Bit biner dari *secret message* ditambahkan 8 bit atau lebih di awal.

Selanjutnya, pada proses pembacaan *cover text*, sistem membaca *cover text* dalam bentuk docx menggunakan Apache POI. *Cover text* yang didapatkan lalu diubah menjadi nilai Unicode. Kemudian, nilai Unicode melalui proses normalisasi untuk menghasilkan posisi rarangken yang sesuai. Normalisasi yang dilakukan mencakup perbaikan posisi rarangken paneleng dan rarangken yang bertumpuk.

Selanjutnya, sistem menghitung jumlah rarangken. Penghitungan ini dipengaruhi oleh *shifting option* yang dipilih sebelumnya. Dalam *shift all*, satu aksara ngalagena yang memiliki satu atau banyak rarangken hanya dihitung sebagai satu rarangken. Dalam *shift group*, satu aksara ngalagena yang memiliki banyak rarangken, dihitung berdasarkan jumlah kelompok posisi rarangken yang ada.

Tabel 1.
Kelompok Posisi Rarangken

Posisi	Rarangken				
Depan					
Belakang					
Atas					
Bawah					

Setelah mendapatkan jumlah rarangken, sistem membandingkannya dengan panjang bit *secret message*. Jika jumlah rarangken lebih dari panjang bit *secret message*, maka sistem menambah bit biner acak di belakang bit *secret message* sesuai selisih jumlah rarangken dan panjang bit *secret message*. Jika jumlah rarangken kurang dari panjang bit *secret message*, maka sistem kembali ke antarmuka pengisian masukan. Jika jumlah rarangken sama dengan panjang bit *secret message*, maka proses pergeseran rarangken dapat dilakukan.

Selanjutnya, proses pergeseran rarangken diatur berdasarkan bit *secret message* dan *shifting option*. Sistem memeriksa setiap nilai Unicode yang ada. Jika termasuk rarangken maka dilakukan pengecekan bit *secret message* yang ada. Jika bit *secret message* bernilai "1", maka rarangken mengalami pergeseran, tetapi jika bit *secret message* bernilai "0", maka rarangken tidak mengalami pergeseran.

Proses pergeseran rarangken dilakukan dengan mengganti huruf yang mewakili rarangken normal pada blok Unicode asal dengan huruf yang mewakili rarangken modifikasi pada blok Unicode baru. Pergeseran yang dilakukan mengacu pada masukan *shifting option*. Sistem menggeser atau tidak semua rarangken dalam satu aksara ngalagena untuk satu bit jika opsi yang dipilih adalah *shift all*. Pada opsi *shift group*, sistem menggeser atau tidak satu kelompok posisi rarangken dalam satu huruf untuk satu bit.

Setelah mendapatkan *stego text*, sistem melakukan proses penyimpanan *stego text*. Proses yang menggunakan Apache PDFBox ini dimulai dengan pembuatan dokumen baru. Setelah itu, sistem menambahkan halaman pada dokumen tersebut. Sebelum memulai menulis *stego text*, sistem menentukan jenis *font* dan ukurannya. Sistem juga menentukan posisi awal yang berada di atas. Jenis *font* yang digunakan adalah *font Sundanese Unicode 2013* hasil modifikasi. Sistem mulai menulis *stego text* dari posisi awal. Ketika *stego text* sudah mencapai batas bawah posisi, maka sistem membuat halaman baru, mengatur ulang *font* dan memulai kembali menulis *stego text* dari posisi atas sampai selesai.

C. Deskripsi Modul Extracting

Modul *extracting* bertujuan untuk mengembalikan *secret message* dari *stego text* yang diberikan. Modul *extracting* menerima data masukan berupa *shifting option* dan *stego text* dalam bentuk PDF. Secara umum terdapat 3 proses dalam

modul *extracting* yaitu pembacaan *stego text*, penghitungan panjang *secret message*, dan ekstraksi *secret message*.

Proses pembacaan *stego text* dimulai dengan membaca data berkas PDF yang berisi *stego text*. Pembacaan ini dilakukan menggunakan bantuan text stripper dari Apache PDFBox. Hasil pembacaan data berupa string aksara Sunda. Hasil ini kemudian diubah menjadi nilai Unicode untuk dilakukan penghitungan panjang *secret message*.

Selanjutnya, proses penghitungan panjang *secret message* dilakukan dengan memeriksa setiap rangkén. Pemeriksaan dipengaruhi oleh *shifting option*. Setiap rangkén diperiksa apakah mengalami pergeseran atau tidak. Rangkén yang mengalami pergeseran menempati blok Unicode modifikasi ketika diperiksa. Jika suatu rangkén *stego text* mengalami pergeseran, maka didapatkan bit panjang *secret message* bernilai “1”. Sebaliknya, jika suatu rangkén tidak mengalami pergeseran, maka bit yang didapatkan bernilai “0”. Proses penghitungan panjang berhenti ketika bit penanda panjang *secret message* yang didapatkan berjumlah sesuai dengan jumlah bit yang telah ditentukan sebelumnya. Dalam hal ini, bit berjumlah 8 bit atau lebih. Jumlah bit yang ditentukan sama pada modul *embedding* maupun *extracting*. Bit panjang *secret message* tersebut dikonversi ke dalam bentuk desimal.

Proses ekstraksi *secret message* dilakukan dengan melanjutkan pemeriksaan setiap rangkén setelah proses penghitungan panjang *secret message*. Setelah didapatkan bit sejumlah panjang *secret message* yang didapatkan pada proses sebelumnya, pemeriksaan rangkén berhenti. Bit *secret message* yang didapatkan pada proses ekstraksi dikonversi ke dalam nilai ASCII untuk setiap 8 bit. Setelah itu, nilai ASCII akan dikembalikan ke dalam bentuk teks *secret message*. Teks tersebut ditampilkan dalam antarmuka program.

III. UJI COBA

A. Data Uji Coba

Data uji coba sistem steganografi teks pada aksara Sunda adalah teks aksara sunda dalam bentuk docx sebagai *cover text*. Hasil uji coba ditunjukkan pada Tabel 2. Ukuran *cover text* dalam bentuk PDF digunakan untuk uji coba perubahan ukuran dan *capacity ratio*. Teks aksara Sunda yang digunakan merupakan teks puisi.

Tabel 2.
Data Uji Coba

No	Nama Berkas	Ukuran docx (kB)	Ukuran pdf (kB)
1	Halodo Teuing ku Panjang.docx [14]	4,79	7,17
2	Itungan Samemeh Balitungan.docx [15]	5,09	8,24
3	Sekar Merdika.docx [16]	4,88	6,92
4	Walungan.docx [17]	5,37	8,48
5	Baris Kahiji.docx [18]	5,25	8,59
6	Pupujian Nabi Urang.docx [19]	8,93	14,78
7	Kumpulan Puisi Godi Suwarna.docx [20]	18,3	19,20
8	Jante Arkidam.docx [21]	5,53	8,96

9	Sajak-sajak Lia Taliah.docx [22]	5,57	9,88
10	Antologi Pasanggiri Maca Sajak Sunda.docx [23]	9,50	16,62

Tabel 3.

Penanda Panjang *Secret Message*

Skenario	Min. Rangkén	Maks. Karakter	Jml. Rangkén
A	16	32	264
B	17	64	521
C	18	128	1034
D	19	256	2059
E	20	512	4108

B. Uji Coba Fungsionalitas

Uji coba fungsionalitas bertujuan untuk mengetahui fungsionalitas yang dapat berjalan pada sistem. Fungsionalitas yang mencakup proses *embedding* dan *extracting* pada *shifting option* yaitu *shift all* dan *shift group* dapat berjalan dengan baik.

C. Uji Coba Panjang *Secret Message*

Uji coba panjang *secret message* dilakukan dengan menghitung minimal rangkén yang diperlukan *cover text* untuk menyisipkan satu karakter, maksimal karakter yang dapat disisipkan dan jumlah rangkén yang diperlukan *cover text* untuk mendapatkan nilai maksimal karakter. Pada uji coba ini, skenario A, B, C, D, dan E mempunyai penanda panjang berturut-turut 8, 9, 10, 11, dan 12 bit. Hasil uji coba ditunjukkan pada Tabel 3 dan 4.

Minimal rangkén yang dibutuhkan *cover text* untuk menyisipkan satu karakter dihitung berdasarkan jumlah bit penanda panjang *secret message* ditambah dengan 8 bit yang merepresentasikan satu karakter. Pada skenario A, bit penanda panjang *secret message* yang berjumlah 8 bit ditambahkan 8 bit representasi satu karakter sehingga minimal rangkén berjumlah 16 bit. Sementara itu, Maksimal karakter dihitung berdasarkan 2^n dibagi 8 dimana n adalah penanda panjang *secret message*. Jumlah rangkén yang dibutuhkan *cover text* untuk menampung maksimal karakter dihitung berdasarkan $2^n + n$ dimana n adalah penanda panjang *secret message*.

Setiap penambahan satu bit penanda panjang *secret message*, terdapat kenaikan 2 kali maksimal karakter yang dapat disisipkan. Semakin tinggi bit penanda panjang *secret message*, maka semakin banyak karakter yang bisa disisipkan dan semakin banyak jumlah rangkén yang harus dimiliki oleh *cover text*.

Tabel 4.

Total Rangkén dan Panjang Karakter

Data	Total Rangkén		Menampung Maksimal				
	Shift All	Shift Group	A	B	C	D	E
1	187	193	X	X	X	X	X
2	188	206	X	X	X	X	X
3	158	168	X	X	X	X	X
4	307	321	V	X	X	X	X
5	244	254	X	X	X	X	X

6	1300	1372	V	V	V	X	X
7	3281	3562	V	V	V	V	X
8	408	434	V	X	X	X	X
9	344	360	V	X	X	X	X
10	1572	1671	V	V	V	X	X

Pada perbandingan yang dilakukan, data ke-4, 8, dan 9 dapat menampung maksimal karakter dari skenario A yaitu 32 karakter. Sementara itu, data ke-6, 7 dan 10 dapat menampung maksimal karakter dari skenario A, B, dan C atau sejumlah 128 karakter. Data ke-7 yang mempunyai rarakngén tertinggi dapat menampung maksimal karakter dari skenario D yaitu 256 karakter. Data ke-3 merupakan data dengan total rarakngén terendah.

D. Uji Coba Perubahan Ukuran Teks

Uji coba perubahan ukuran bertujuan untuk mengetahui perubahan ukuran dari *cover text* menjadi *stego text*. Uji coba perubahan ukuran dihitung dengan membandingkan ukuran *cover text* dan *stego text* yang dihasilkan untuk setiap skenario. Grafik hasil uji coba ditunjukkan pada Gambar 1.

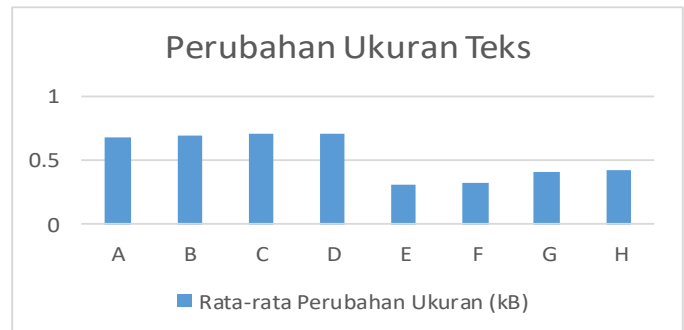
Uji coba ini memiliki 8 skenario yaitu A, B, C, D, E, F, G, H. Skenario A memiliki masukan *secret message* “rijal” menggunakan pergeseran *shift all* dan sisa rarakngén digeser secara acak. Skenario B memiliki masukan *secret message* “rijal” menggunakan pergeseran *shift group* dan sisa rarakngén digeser secara acak. Skenario C memiliki masukan *secret message* “nama saya rijal” menggunakan pergeseran *shift all* dan sisa rarakngén digeser secara acak. Skenario D memiliki masukan *secret message* “nama saya rijal” menggunakan pergeseran *shift group* dan sisa rarakngén digeser secara acak. Skenario E memiliki masukan *secret message* “rijal” menggunakan pergeseran *shift all* dan sisa rarakngén tidak digeser secara acak. Skenario F memiliki masukan *secret message* “rijal” menggunakan pergeseran *shift group* dan sisa rarakngén tidak digeser secara acak. Skenario G memiliki masukan *secret message* “nama saya rijal” menggunakan pergeseran *shift all* dan sisa rarakngén tidak digeser secara acak. Skenario H memiliki masukan *secret message* “nama saya rijal” menggunakan pergeseran *shift group* dan sisa rarakngén tidak digeser secara acak.

Untuk skenario A, B, C, dan D dilakukan 3 kali percobaan karena skenario tersebut menggunakan algoritma pergeseran acak sisa rarakngén. Hasil 3 kali percobaan tersebut kemudian didapatkan rata-ratanya. Skenario E, F, G dan H hanya dilakukan 1 kali percobaan karena menghasilkan nilai yang tetap.

Secara umum, skenario yang tidak menggunakan algoritma pergeseran acak sisa rarakngén yaitu skenario E, F, G, dan H memiliki perubahan ukuran yang lebih kecil yaitu 0,37 kiloByte. Sementara itu skenario A, B, C, dan D yang menggunakan algoritma pergeseran acak sisa rarakngén, memiliki perubahan sebesar 0,70 kiloByte. Terdapat penurunan sebesar 47,14% jika algoritma pergeseran acak sisa rarakngén dimatikan.

Dari uji coba yang dilakukan, pilihan *shifting option* tidak terlalu berpengaruh pada perubahan ukuran pada *stego text* yang dihasilkan. Masukan *secret message* pada uji coba ini secara umum berpengaruh pada perubahan ukuran teks. Pada skenario E dan F, terdapat perubahan teks sebesar 0,32

kiloByte untuk penyisipan 5 karakter *secret message*. Sementara itu pada skenario G dan H, terdapat perubahan teks sebesar 0,41 kiloByte untuk penyisipan 15 karakter *secret message*.



Gambar 1. Grafik Perubahan Ukuran Teks

E. Uji Coba Waktu

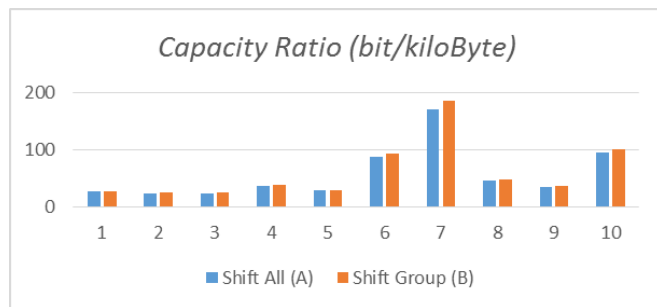
Uji coba waktu bertujuan untuk mengetahui waktu yang diperlukan untuk melakukan proses *embedding* dan *extracting*. Uji coba ini mempunyai 8 skenario yang sama dengan uji coba perubahan ukuran teks.

Uji coba waktu *embedding* dihitung ketika proses pergeseran rarakngén dijalankan. Secara umum, skenario A, B, C, dan D yang menggunakan algoritma pergeseran acak sisa rarakngén memiliki waktu *embedding* 89,24 *miliseconds*. Sementara itu, skenario E, F, G dan H yang tidak menggunakan pergeseran acak memiliki waktu *embedding* yang lebih kecil yaitu 79,13 *miliseconds*. Terdapat penurunan 11,32% ketika algoritma pergeseran acak sisa rarakngén dimatikan.

Uji coba waktu *extracting* dihitung ketika proses ekstraksi informasi dijalankan. Secara umum, skenario A, B, E dan F yang memiliki masukan 5 karakter *secret message* membutuhkan waktu 402,87 *miliseconds*, sedangkan C, D, G, dan H yang memiliki masukan 15 karakter *secret message* membutuhkan waktu 1204,54 *miliseconds*. Terdapat kenaikan 198,99% jika karakter yang menjadi masukan bertambah 10 karakter *secret message*. Tidak ada perbedaan signifikan pada penggunaan algoritma pergeseran acak sisa rarakngén atau pemilihan *shifting option*.

F. Uji Coba Capacity Ratio

Uji coba *capacity ratio* dihitung dengan membagi total rarakngén dengan ukuran *cover text*. Besaran hasil *capacity ratio* adalah *bit/kiloByte*. Uji coba *capacity ratio* memiliki dua skenario yaitu A dengan pilihan *shift all* dan B dengan pilihan *shift group*. Dari uji coba yang dilakukan, data ke-6, 7 dan 10 memiliki *capacity ratio* yang lebih tinggi dibandingkan data lainnya. Hal tersebut disebabkan total rarakngén yang nilainya mencapai lebih dari 1000 rarakngén.



Gambar 2. Grafik Capacity Ratio

Secara umum, skenario B yang menggunakan *shift group* memiliki *capacity ratio* yang lebih tinggi disebabkan total rangkén yang lebih banyak. Terdapat kenaikan 6,54% dari semula 57,01 *bit/kiloByte* pada skenario A dengan *shift all* menjadi 60,74 *bit/kiloByte* pada skenario B dengan *shift group*.

IV. KESIMPULAN/RINGKASAN

Kesimpulan yang diperoleh dari hasil uji coba dan evaluasi penelitian ini adalah sebagai berikut:

1. Steganografi teks pada aksara Sunda dengan pendekatan Feature Coding dilakukan dengan menggeser rangkén sesuai pilihan pergeseran (*shifting option*) yang ada. Pergeseran dilakukan dengan mengganti blok Unicode rangkén normal menjadi blok Unicode rangkén modifikasi.
2. Maksimal karakter yang dapat disisipkan pada *cover text* dipengaruhi oleh bit penanda panjang *secret message* dan kapasitas *cover text*. Ukuran *stego text* yang dihasilkan dipengaruhi oleh masukan *secret message* dan algoritma pengacakan sisa rangkén. Waktu *embedding* secara umum dipengaruhi oleh penggunaan algoritma pergeseran acak sisa rangkén. Waktu *extracting* dipengaruhi oleh panjang masukan *secret message*.
3. Capacity Ratio dari sebuah *cover text* dalam aksara Sunda dipengaruhi oleh jumlah rangkén yang terdapat pada *cover text* tersebut. Pilihan *shift all* menghasilkan rata-rata *capacity ratio* 57,01 *bit/kiloByte*. Pilihan *shift group* menghasilkan rata-rata *capacity ratio* yang lebih besar yaitu 60,74 *bit/kiloByte*. Terdapat kenaikan sebesar 6,54%.

UCAPAN TERIMA KASIH

Penulis M.R mengucapkan terima kasih kepada Ilham Nurwansah yang telah memberikan penjelasan seputar aksara Sunda dan Unicode.

DAFTAR PUSTAKA

- [1] S. Changder, D. Ghosh and N. C. Debnath, "Linguistic approach for text steganography through Indian text," in 2nd Int. Conf. on Computer Technology and Development, 2010.
- [2] D. Kahn, The Code Breakers- the comprehensive history of secret, Schribner, 1996.
- [3] J. T. Brassil, S. Low, N. F. Maxemchuk and O. L., "Electronic Marking and Identification Techniques to Discourage Document Copying," IEEE Journal on Selected Areas in Communications, vol. 13, no. 8, pp. 1495-1504, 1995.

- [4] M. H. Shirali-Shahreza and M. Shirali-Shahreza, "A new approach to persian/arabic text steganography," in 5th Int. Conf. Computer and Information Science, Washington, 2006.
- [5] A. Gutub and M. Fattani, "A Novel Arabic Text Steganography Method," in WASET International Conference on Computer., Vienna, 2007.
- [6] M. Aabed, S. Awaideh, A. Elshafei and A. Gutub, "Arabic Diacritics Based Steganography," in IEEE International Conference on Signal Processing and Communications (ICSPC'07), 2007.
- [7] A. Odeh, K. Elleithy and M. Faezipour, "Steganography in Arabic text using Kashida variation algorithm (KVA)," in Systems, Applications and Technology Conference (LISAT), Long Island, 2013.
- [8] A. Odeh, A. Alzubi, Q. B. Hani and K. Elleithy, "Steganography by multipoint Arabic letters," in Systems, Applications and Technology Conference (LISAT), Long Island, 2012.
- [9] K. Alla and R. S. R. Prasad, "A Novel Hindi Text Steganography using Diacritics and its Compound words," International Journal of Computer Science and Network Security (IJCSN), vol. 8, no. 12, pp. 404-408, 2008.
- [10] S. Changder, N. C. Debnath and D. Ghosh, "Hindi Text Steganography by Shifting of Matra," in International Conference on Advances in Recent Technologies in Communication and Computing, 2009.
- [11] K. Alla and R. S. R. Prasad, "A New Approach to Hindi Text Steganography Using Matraye, Core Classification And HHK Scheme," in Seventh International Conference on Information Technology, 2010.
- [12] K. Alla and R. S. R. Prasad, "A new approach to Telugu text steganography," in IEEE Symposium on Wireless Technology and Applications (ISWTA), 2011.
- [13] Tim Unicode Aksara Sunda 2008, Direktori Aksara Sunda untuk Unicode, Pemerintah Provinsi Jawa Barat, 2008.
- [14] Y. Hendayana, "Sajak Bahasa Sunda (Halodo Teuing Ku Panjang)," bahasa-sunda.com, [Online]. Available: <http://bahasa-sunda.com/contoh-sunda/sajak-bahasa-sunda/sajak-bahasa-sunda-halodo-teuing-ku-panjang.html>.
- [15] I. Nugraha, "Manglé - Majalah Mingguan Basa Sunda | Itungan Saméméh Balitungan," Manglé, [Online]. Available: http://mangle-online.com/pidangan/rubrik/sjk/Manglé%20-%20Majalah%20Mingguan%20Basa%20Sunda%20_%20Itungan%20Saméméh%20Balitungan.htm.
- [16] Risnawati, "Manglé - Majalah Mingguan Basa Sunda | Sekar Merdika," Manglé, [Online]. Available: http://mangle-online.com/pidangan/rubrik/sjk/Manglé%20-%20Majalah%20Mingguan%20Basa%20Sunda%20_%20Sekar%20Merdika.htm.
- [17] U. Romli HM, "Manglé - Majalah Mingguan Basa Sunda | Walungan," Manglé, [Online]. Available: http://mangle-online.com/pidangan/rubrik/sjk/Manglé%20-%20Majalah%20Mingguan%20Basa%20Sunda%20_%20Walungan.htm.
- [18] C. Nurzaman, "Manglé - Majalah Mingguan Basa Sunda | Baris Kahiji," Manglé, [Online]. Available: http://mangle-online.com/pidangan/rubrik/sjk/Manglé%20-%20Majalah%20Mingguan%20Basa%20Sunda%20_%20Baris%20Kahiji.htm.
- [19] A. Karsono and C. Retty Isnendes, Lir Cahya Nyorot Eunteung, Bandung: Sonagar Press, 2008.
- [20] G. Suwarna, "Kumpulan Sajak-Sajak Godi Suwarna," cikancah-cyber, [Online]. Available: <https://www.cikancah-cyber.com/2016/09/kumpulan-sajak-sajak-godi-suwarna.html>.
- [21] A. Rosidi, "Sajak Bahasa Sunda (Jante Arkidam)," bahasa-sunda.com, [Online]. Available: <https://bahasa-sunda.com/contoh-sunda/sajak-bahasa-sunda/sajak-bahasa-sunda-jante-arkidam.html>.

- [22] L. Taliah, "Sajak-sajak Lia Taliah," sundanews.com, [Online]. Available: <http://www.sundanews.com/aos/seratan/1352485842>.
- [23] SMA Pasundan 2 Tasikmalaya, ANTOLOGI PASANGGIRI MACA SAJAK SUNDA XIV, Tasikmalaya: SMA Pasundan 2 Tasikmalaya, 2015.